Incomplete Contracts and Complexity Costs

Luca Anderlini and Leonardo Felli

London School of Economics

1998

INCOMPLETE CONTRACTS AND COMPLEXITY COSTS*

LUCA ANDERLINI
(St. John's College, Cambridge)

LEONARDO FELLI
(London School of Economics)

January 1997

ABSTRACT. This paper investigates, in a simple risk-sharing framework, to which extent the incompleteness of contracts could be attributed to the *complexity costs* associated with the writing and the implementation of contracts. We show that, given any measure of complexity in a very general class, it is possible to find simple contracting problems such that, when complexity costs are explicitly taken into account, the contracting parties optimally choose an *incomplete* contract which coincides with the 'default' division of surplus. Optimal contracts with complexity costs are constrained efficient in our model. We therefore interpret our results as saying that, in the absence of a *strategic* role for complexity costs, their effect is entirely determined by their size relative to the size of payoffs.

JEL CLASSIFICATION: C70, C78, D80, D81.

KEYWORDS: Incomplete Contracts, Complexity Measures, Computability.

ADDRESS FOR CORRESPONDENCE: Leonardo Felli, The London School of Economics & Political Science, Department of Economics, Houghton Street, London WC2A 2AE, England.

*We wish to thank an anonymous referee, Ed Green, John Moore, Ariel Rubinstein, Oved Yosha, and seminar participants at Cambridge, LSE, Tel-Aviv, Hebrew University of Jerusalem, the 1995 Winter Symposium of the Econometric Society and the European Summer Symposium in Economic Theory in Guerzensee for insightful comments. Financial support from the Suntory and Toyota International Centres for Economics and Related Disciplines is gratefully acknowledged. We are solely responsible of any remaining errors.

1. INTRODUCTION

1.1. Overview

The contracts which economic agents write to regulate their transactions are often *incomplete* in the sense that they take into account 'less information' than would be optimal for the agents to include. One often cited reason for the incompleteness of contracts is the presence of *complexity costs* associated with writing and implementing contracts (Grossman and Hart 1986, Hart and Moore 1988).

This paper investigates the role of complexity costs in a simple risk-sharing model. We show that complexity costs may lead to contract incompleteness. However, in our model, complexity costs do not play a genuinely strategic role. This leads to a choice of contract, after complexity costs are taken into account, which is *constrained efficient*. We interpret this as saying that in the absence of a strategic role, the effect of complexity costs is entirely determined by their size, relative to the share of the surplus which the contract allocates to each party. In Anderlini and Felli (1997), we investigate a model in which *ex-ante* contractual costs play a strategic role akin to that of 'relationship-specific investments' in a 'hold-up' problem (Hart and Moore 1988).

We restrict ourselves to contracting problems involving two agents only. The two parties face today an environment in which uncertainty is embodied in the resolution of a state of nature tomorrow. They may decide to specify today a 'sharing rule' which prescribes a pair of 'outcomes'—one for each agent—as a function of the particular state which will be realized tomorrow. We do not consider the possibility of contracting over actions and/or any informational asymmetries, although we believe that our approach to the problem naturally extends to cover some of these cases.

We restrict ourselves to contracts which can be viewed as a well specified 'procedure' consisting of a finite set of 'clauses' which embody the agents' chosen sharing rule in the sense that once the state of nature is realized, it can be used as an 'input' of what is specified in the contract which yields the prescribed pair of outcomes in a finite number of 'steps'. In Anderlini and Felli (1994), we argued at length that this restriction captures the nature of what agents can and cannot do when they are restricted to choose *written* contracts. We do not repeat the arguments here; the same interpretation applies unchanged to the analysis we carry out in this paper.

Once a contract is viewed as a procedure as we have just described, it is immediately clear that we can distinguish between two types of complexity costs associated with it, which for the time being we simply call 'ex-ante' and 'ex-post'.[1] Ex-ante complexity costs are those associated with defining the procedure. For written contracts these are the costs of writing out in full the text of the contract itself. Ex-post complexity costs are those associated with 'working out' the outcomes which the procedure prescribes once the state of nature is realized. They can be viewed as embodying both the pure 'computational' costs associated with a particular contract-state pair and the costs of actually implementing the outcomes which the contract prescribes in each particular state.

Viewing a contract as a procedure in the above way does not of course pin down the form of the complexity cost functions in any particular way. Indeed, we take the view that the exact form of the complexity costs associated with writing and implementing a contract depends on an extremely large (and complex) set of circumstances which define the contract problem at hand. Several elements which may affect what is difficult and what is easy to do with a contract immediately come to mind.

Some of the factors affecting the complexity costs associated with a contract could be termed 'environmental'. These are the characteristics of the larger system in which a contractual arrangement is embedded. The language in which a contract is drafted may make a difference. For instance, a contract concerning computer software is probably 'easier' to write in English than in Classical Arabic. On the other hand a contract which concerns observance of certain Islamic religious rules is very probably more easily written in Arabic than in English.

Any contract is 'embedded' in a larger legal system, which prescribes a variety of formalities, for instance, in order for the contract to be valid and enforceable by a court. Some legal systems may allow certain transactions easily, while others may even explicitly forbid them. An example are lawyers' contingency fees which are the norm in the Unites States, but (up to 1994) were forbidden in the United Kingdom. A further example is the procedure to change a person's name. In the United Kingdom this is extremely simple (deed poll), while in some Roman-Napoleonic legal systems, it is an extremely complex matter. In Italy, for instance, it requires a judgement of the Court of Appeal.

A second variety of factors affecting the complexity costs associated with a contract are specific to the particular situation at hand. Consider, for instance, a co-insurance problem in which the parties' *property rights* over the state-contingent surplus are defined in a particular way. Clearly, a contract which prescribes that each party keeps the portion of surplus over which they have property rights is simpler to write and implement (in this extreme case it is likely that writing no contract at all will suffice) than a contract which prescribes some state-contingent transfers between the agents. Indeed, the fact that to a contracting problem there is usually associated some 'default' sharing rule which requires minimal complexity costs to write and implement, figures explicitly in our model below.

To take into account the fact that a variety of complexity costs are possible, we consider a wide class of possible complexity cost functions in our analysis below. Another possible way of proceeding (as in Dye (1985)) is, of course, to try to specify fully a particular complexity measure which 'fits well' the contractual problem at hand. The advantage of this way to go is that a fuller characterization of the equilibrium contracts is possible. The disadvantages, in our view, are two-fold. First of all, as we argued above, it may be very difficult to specify a particular complexity measure which will accurately reflect all the factors which determine what is complex and what is not in a given contractual problem: some element of arbitrariness will always be present. Secondly, as we argue below (Subsection 1.2), the results obtained in this way will not be robust to a change in either the contracting problem, or in the factors which determine which one is the 'correct' complexity measure to use.

Since we are not willing to specify a precise form of the complexity cost function, our results are necessarily of the type "given any complexity cost function in a particular class, the chosen contract has characteristic x or y...", and the feature of the contract we choose to concentrate on is *incompleteness*.

To keep the analysis as simple as possible, we focus on an extreme form of incompleteness. We call incomplete a contract that does not modify the default allocation of surplus (provided that the default allocation is not the *first best*). Clearly our definition of incompleteness is then 'too narrow'. It suffices for our analysis below, however. Moreover, it is compatible with every definition of incompleteness found in the existing literature (Grossman and Hart 1986, Hart and Moore 1988, Anderlini

and Felli 1994, Tirole 1994, among others).

The main result of our analysis is that, given any complexity cost function in a very general class, one can find a contracting problem such that, when complexity costs are taken into account, the parties will choose an *incomplete* contract. Indeed for any given complexity cost function it is always possible to find a co-insurance problem 'complex' enough to make it not worthwhile for the parties to write any contract. We then proceed to show that, as the relative size of the complexity costs shrinks to zero, the limit optimal contract given complexity costs coincides with the first best allocation of surplus.

The plan of the paper is as follows. The class of contracting problems which we consider is introduced in Section 2. Section 3 presents the notion of an algorithmic or computable contract. We then proceed to describe the class of complexity cost functions which we associate to computable contracts (Section 4) and the parties' decision problem when complexity costs are taken explicitly into account (Section 5). The main results of the paper are contained in Section 6 in which we partly characterize optimal computable contracts with complexity costs. Section 7 concludes the paper. Before proceeding any further, we briefly discuss some related literature.

1.2. Related Literature

Starting from Grossman and Hart (1986) a number of papers have discussed incomplete contracts. Most of these papers, however, assume that contracts are incomplete and concentrate on the role of available mechanisms, and in particular institutions, in mitigating the inefficiencies generated by contract incompleteness. Some of the mechanisms considered are: vertical and lateral integration (Grossman and Hart 1986), the optimal allocation of ownership rights on physical capital (Hart and Moore 1990), and the delegation of authority within organizations (Aghion and Tirole 1997). We differ from these papers since we do not assume contractual incompleteness but rather derive it *endogenously* from the complexity costs associated with the writing and implementing of a contract.

A number of recent papers have asked the question of why contracts are incomplete. Hart and Moore (1988) and a number of subsequent papers (Chung 1991, Aghion, Dewatripont and Rey 1994, Nöldeke and Schmidt 1995) have explored

whether one of the causes of contractual incompleteness is the fact that the outcome that the parties wish to implement through a contract may be, at least in part, unobservable by the enforcing agency (the court). They conclude that there exist circumstances in which the parties will write a contract which is silent on certain (or all) states. Such contract will leave out some details that the court cannot observe. However, in a recent paper Maskin and Tirole (1996) show that in such an environment it is possible to devise a mechanism that implements the same outcome that a complete contract would implement. This is achieved by asking the contracting parties—once the relevant state of nature is realized—to report such a state or, equivalently, to report the utility levels accruing to each party in such a state and inducing through incentives truthful revelation. By contrast, we assume that the court can observe the realized state of nature but there exist costs associated with describing and implementing the mechanism the parties would like the court to implement.

The papers most closely related to the present one are Dye (1985) and Anderlini and Felli (1994). Dye (1985), in a contractual set-up which is different from the one we analyse below, postulates a complexity cost function which is increasing in the 'number of contingencies' which a contract takes into account (the number of 'cells' into which the state space is partitioned by the contract). He then proceeds to derive a parametric class of equilibrium contracts which are incomplete. However, the specific complexity measure which he postulates leads to some un-appealing results. For instance, Hart and Holmström (1987) argue that if the optimal contract is, say, linear in the random size of the surplus, then it will have an infinite complexity cost and therefore will never be chosen. However, if the same contract is written as shares of the surplus accruing to the parties as a function of the state of nature, then its complexity cost will be zero. This type of criticism does not apply to our analysis below since we obtain results which are valid for any complexity measure in a very general class.

In Anderlini and Felli (1994) we use the same model of a contract as an algorithmic procedure (a Turing Machine) used in this paper to explore whether this restriction may explain, by itself, contractual incompleteness. We conclude that restricting the parties to choose algorithmic contracts has a negligible impact on the parties' expected utilities unless the restriction is paired with an equivalent restriction on the

parties' decision process. Hence, in the absence of explicit bounds on the parties' ability to select or write and enforce algorithmic contracts, every feasible contract may be 'approximated' by an algorithmic one. In this paper we go one step further. We impose explicit complexity bounds on the parties' ability to write and enforce computable contracts and inquire whether the resulting outcome may be meaningfully defined to be an incomplete contract.

In a recent paper, Segal (1995) focuses on a contracting problem in which the relevant state of nature is not observable to the enforcing agency (the court). In such an environment he analyzes the parties' welfare gains from using message contingent mechanisms as in (for instance) Maskin and Tirole (1996). He shows that such gains become negligible as the number of relevant states of nature increases without bound. The implication is then that message contingent mechanisms will not be used if they entail a complexity cost. As well as in the formulation of the contracting problem, we differ from Segal (1995) in that we explicitly introduce complexity costs.

Finally, starting from Rubinstein (1986) a literature has developed that explores the impact of complexity considerations in repeated strategic interaction (Abreu and Rubinstein 1988, Piccione 1992, Rubinstein and Piccione 1993). We differ from this literature in the complexity measure we choose as well as in the nature of the model. Indeed, this literature focuses on the complexity measure associated with the computing device strategically chosen by the players. We associate complexity with the computations performed by a computing device (a contract), rather than with the computing device itself. In our framework, a 'complex' computing device could be used to perform a 'simple' computation, or viceversa. It is the complexity of the actual computation and not of the device which matters in our model.

2. THE CONTRACTING PROBLEM

We consider a general characterization of a simple co-insurance problem. Two risk averse parties agree *ex-ante* on a contract which allows them to share the risk associated to the common random environment in which they operate.

Let $\mathcal{S} = \{1, \ldots, s, \ldots, N\}$ be a finite set of mutually exclusive possible states of nature, and $P = \{p_1, \ldots, p_N\}$ a probability distribution over $\mathcal{S}$. We take π_s to denote the size of the 'surplus' jointly available to the two contracting agents as a function

of the realized state of nature. We assume that, in the absence of any contractual prescription, the *property rights* of the two parties over the surplus are well defined. This will identify the default outcome which we mentioned in the Introduction. Let d_s^i (with $d_s^i \geq 0$) be the share of the surplus owned by agent $i = 1, 2$ when the state of nature is s. We also set $d_s^1 + d_s^2 = \pi_s$, for any $s \in \mathcal{S}$, and denote by d_s the pair (d_s^1, d_s^2) and by d^i the vector $(d_1^i, \ldots, d_N^i)$.

A contract can now be viewed, without loss of generality, as an object which, as a function of the state, indicates how much of the surplus should go to each agent. The 'outcome' of a contract in state s is the pair $r_s = (r_s^1, r_s^2)$ specifying the pair of shares accruing to each party in state s. The entire contract is denoted by $r = (r_1, \ldots, r_N)$. We restrict contracts to satisfy $r_s^i \geq 0$ for $i = 1, 2$ and for all $s \in \mathcal{S}$. We are therefore assuming that some form of limited liability constrains the contracts which the parties can choose. Finally, it is natural to assume that $r_s^1 + r_s^2 \leq d_s^1 + d_s^2 = \pi_s$ for all $s \in \mathcal{S}$.

Given a sharing rule r let agent i's preferences in state s be represented by the following state contingent utility function denoted by $U_i(r_s^i, s)$. For each $s \in \mathcal{S}$ we assume $U_i(\cdot, \cdot)$ to be strictly increasing, continuous and concave in its first argument. To avoid corner solutions we also assume $\lim_{x \to 0} \partial U_i(x, s)/\partial x = +\infty$ for all $s \in \mathcal{S}$.

Similarly to Anderlini and Felli (1994), we consider a general formulation of the bargaining procedure which yields the optimal contract between the two agents. We take the 'reduced form' of the bargaining procedure to be a function $G : \mathbb{R}^2 \to \mathbb{R}$ which maps the two parties' expected utility levels into the reals. We take G to be continuous and increasing in both arguments. Let

$$\mathcal{G}[r] \equiv G\left\{\sum_{s=0}^{N} U_1(r_s^1, s)p_s, \sum_{s=0}^{N} U_2(r_s^2, s)p_s\right\} \tag{1}$$

The default sharing rule we have defined above clearly yields a lower bound on the expected utility which the agents can achieve by 'walking out' of the contractual bargaining. We therefore set agent i's reservation expected utility level to be $\sum_{s=1}^{N} U_i(d_s^2, s)p_s$.

We have now all the elements to define the set of *first best* sharing rules as the set

R^* of all r^* which solve the following problem:

$$\begin{aligned} \max_{r} \quad & \mathcal{G}[r] \\ s.t. \quad & r \in \mathcal{F} \end{aligned} \tag{2}$$

where $\mathcal{F}$ denotes the set of sharing rules satisfying:

$$\begin{aligned} \sum_{s=1}^{N} U_1(r_s^1, s)p_s \;&\geq\; \sum_{s=1}^{N} U_1(d_s^1, s)p_s \\ \sum_{s=1}^{N} U_2(r_s^2, s)p_s \;&\geq\; \sum_{s=1}^{N} U_2(d_s^2, s)p_s \\ r_s^i \geq 0 \quad \forall\, i = 1,2 \quad & \forall\, s \in \mathcal{S} \\ r_s^1 + r_s^2 \leq d_s^1 + d_s^2 \quad & \forall\, s \in \mathcal{S}. \end{aligned} \tag{3}$$

3. COMPUTABLE CONTRACTS

The contract the parties choose to share the risk is effective only in the event that it can be enforced. We focus on situations in which this is possible only if the contract is a written agreement between the parties. In many economic situations legal restrictions, common practice or simply the contracting parties' will, impose such restriction on enforceable contracts, here we take it simply as given.

The required written form of contracts has consequences as far as the nature of the enforcement mechanism of a contract is concerned. Indeed, we shall assume that the written contract contains all the enforcement prescriptions which the court must apply. In other words, the written contract is a procedure embodying both the prescription associated with every realized state of nature and how these prescriptions must be enforced. In this sense the court that implements a given contract is just a *passive* subject: it simply ensures that the prescriptions of the contract are complied with by the contracting parties.

We view a written contract as a finite set of clauses which, given a state of nature, yields a value of the sharing rule in a finite number of 'steps'. In our context, a finite number of steps may be interpreted as the fact that, examining the contract for a given realized state of nature it is possible to identify the associated sharing rule in finite time. As in Anderlini and Felli (1994) we model the finite set of clauses that associates a value of the sharing rule to a realized state of nature in finite time as

an *algorithmic* function. We adopt what is widely agreed to be the widest possible notion of algorithmic function—a function is algorithmic if and only if it is computable by a Turing machine (or by an abstract computing device in an equivalent class). Throughout the rest of the paper we use the terms algorithmic, Turing-computable, or simply computable in an equivalent way. In essence, we identify a written contract with a Turing machine which computes the values of the sharing rule as a function of the state of nature and of the value of the default in the given state.

A Turing machine is an abstract computing device which performs computations according to a given *programme*. We do not give a formal definition here, but refer instead to standard texts such as Rogers (1967), Cutland (1980), or for a brief exposition Anderlini (1989). Here we only recall that using a standard technique, known as *Gödel numbering*, it is possible to put Turing machines, in a one-to-one (computable) correspondence with the natural numbers. In other words, it is possible to code every computable contract by means of a natural number. Further, the same procedure allows us to code with natural numbers the inputs and outputs of Turing machines. We denote by $\{x\}(e)$ the result of the computation of Turing machine $x \in \mathbb{N}$ on input $e \in \mathbb{N}$. The symbols $\{x\}(e) \uparrow$ and $\{x\}(e) \downarrow$ will denote a computation which halts and does not halt respectively; while the symbol $\{x\}(e_1, \ldots, e_m)$ denotes a Turing machine x applied to the m inputs $e_1, \ldots, e_m$. A *total* computable function is a computable function which is defined for all possible inputs.

Throughout the paper, the symbol $\langle x_1, \ldots, x_n \rangle$ denotes the code in $\mathbb{N}$ of the n-tuple of natural numbers $(x_1, \ldots, x_n)$. If $z \in \mathbb{N}$ is the code of an n-tuple of natural numbers, the notation $\rangle z \langle$ stands for the actual set of numbers coded by z. Notice that our notation, for instance, implies that given any n-tuple $(x_1, \ldots, x_n) \in \mathbb{N}^n$ we have that $\rangle \langle x_1, \ldots, x_n \rangle \langle = (x_1, \ldots, x_n)$.

Using the same symbol as for the contract array of Section 2, a typical computable contract will be a Turing machine $r \in \mathbb{N}$, which takes as inputs the state $s \in \mathcal{S}$, and the default allocation in state s, d_s. Therefore in state s the value of the sharing rule computed by r is given by $\{r\}(s, d_s)$. It should be noted that we take the values of the default allocation for any given state of nature to be a pair of rational numbers, $(d_s^1, d_s^2) \in \mathcal{Q}^2$ so that they can be appropriately coded as an input to the Turing machine r.[2] Indeed, we use the same symbol d_s to denote the code $d_s = \langle d_s^1, d_s^2 \rangle$ and

the actual pair (d_s^1, d_s^2), so as to minimize notation. We also take the shares specified by a computable contract to be rational numbers. In other words, we take the output of $\{r\}(s, d_s)$ to be a pair of rational numbers, $\{r\}(s, d_s) = (r_s^1, r_s^2) \in \mathcal{Q}^2$.

We conclude this section with an observation which clarifies the role of the assumption of computability of contracts in our analysis. It can be shown that in the present context, before complexity costs are taken into account, the assumption that a contract be computable imposes 'negligible' constraints on the agents in their choice of sharing rules. In Anderlini and Felli (1994) we show formally that, provided certain *continuity* conditions are satisfied, then the level of expected utility which the parties can achieve in the first best can always be *approximated* using a computable contract.[3] In this sense, computability of the sharing rule alone imposes negligible constraints on the agents' choices.

In essence the assumption of computable contracts in our analysis below is a device which enables us to define the complexity costs associated with a contract in each state of nature. If we want to identify the source of these costs with the complexity of the operations which the contract prescribes in a given state, we are implicitly assuming that a contract can be identified with a well defined (finite) set of operations in each given state. In other words we are implicitly assuming that the contract is computable in the sense specified above.

4. COMPLEXITY

In the Introduction we argued at length that it is not possible to specify a complexity cost function without reference to a whole host of factors which vary widely from one contract problem to another. We therefore work with an extremely wide class of possible complexity cost functions.

We model complexity costs as being a function of the contract-state-default value triple. This formulation is sufficiently general so that the same function can contain both writing (ex-ante) and implementation (ex-post) costs. The ex-ante costs can be included simply as a term which behaves like a standard fixed cost. The part of the complexity cost function which 'varies' with the state represents implementation costs.

Although the class we consider is large enough to contain functions which are

widely different from each other, we think it is useful to introduce the type of complexity functions we have in mind by means of an analogy. Imagine first of all, that the writing and implementation of the contract in question is carried out by a single unit, which we will call a 'law firm'. This is purely for expositional purposes as it is clearly the case that writing a contract is a lawyer's job, while enforcement often requires the intervention of a third party, frequently a court.

Everyone in the law firm dealing with the contract charges for their time. They do so by filling in 'time sheets' every time they do anything that concerns the contract. The total complexity cost associated with a particular contract-state-default triple is then computed adding up the costs of the various time sheets accumulated in the law firm during the entire process of writing and implementation. (Different types of work can of course be charged at different rates.) This way of computing complexity costs gives us almost automatically four properties of the complexity cost function.

The first is trivial, and it is simply that if the prescription of the contract for a given state-default pair is undefined because the contract requires say to do an infinite number of operations, then the complexity costs are themselves undefined. This could be the case, for instance, with a contract in which clause α calls on clause β to be checked and clause β calls back on clause α in a 'loop.'

The second property is that it is always possible in principle, given a contract and a state-default pair, to work out whether the complexity costs will exceed a particular, arbitrarily given, value. All one has to do is let the law firm start work on the contract-state-default triple, and then monitor the cumulative value of the time sheets through time. If the threshold value is reached before the work is finished, then the complexity costs exceed the given value. If the work ends before the threshold value is reached, then we will know that the complexity costs are below it. Notice that this property holds also for contracts and state-default pairs for which the prescription of the contract is not defined.

The third property is that, given an arbitrary cumulative value for the time sheets again, it is possible to ask and answer the question: what stage of the process does the law firm get to if we stop its operations precisely when the complexity costs reach the given threshold value? (If the value is never reached we take the answer to be the outcome of the contract.) This can clearly be done in much the same way

as for the previous property. We let the firm start the work on the contract-state-default triple. When the threshold value is reached we can imagine stopping all work and collecting the 'interim results' that were reached up to that point. They may constitute a 'coherent' outcome or not, but they can certainly be put together to form a description of the stage that was reached in the work of the law firm.

The fourth property is apparent once we think of the law firm as an organization which has to proceed by 'finite increments' in its work. Suppose that we stopped the operations of the law firm at a particular cumulative value of the time sheets as above. Let this cumulative value be n. Let also θ_n represent the state of the work carried out by the law firm up to then. Suppose that we are also given the specification of the contract-state-default triple. Then, given these four elements, there is only a *finite set* of values which θ_{n+1} can take. In other words, between 'stages' n and $n+1$ only a finite number of operations can be carried out by the law firm.

Notice that the latter property is a statement concerning the set of possible operations that can be performed in a unit of time (between two consecutive stages) within the procedure that implements the computable contract in question. Once the computable contract is chosen by the contracting parties the set of operations necessary to implement that contract is necessarily finite. This is due to the computability restriction which requires the procedure embodied in the contract to deliver a value of the share accruing to each party in finite time. Hence, this property should not be interpreted as a restriction on the set of feasible operations among which the contracting parties may choose when identifying the optimal computable contract. Such a set may indeed be infinite and the property in question still hold.

Once we introduce complexity costs the question arises as to how such costs will be shared by the parties. In general, this sharing will be specified by the contract and it may itself be complex. Hence, a general complexity measure has to include this extra source of complexity. In other words, the complexity measure will depend on how the parties share the complexity costs. In terms of the activity of the law firm described above the cumulated time sheets at every instant of time have to include also the time required by the law firm to handle the part of the contract that stipulates how the contracting parties will share the payment of the complexity costs. Throughout the rest of the paper, we shall denote with k_s^i the share of the complexity costs that

accrues to party i in state of nature s and take $k_s^1 + k_s^2 = 1$ for every state $s \in \mathcal{S}$. Further, for sake of simplicity, we take the shares to be rational numbers $k_s^i \in \mathcal{Q}$ and denote by k_s the pair (k_s^1, k_s^2), and by $k = \langle k_1, \ldots, k_N \rangle$ the code of both parties' shares in every state of nature s.

The class of complexity cost functions which we consider below is the class of all complexity cost functions which satisfy the formal counterparts of the four properties we have just described. Formally, a complexity cost function $c \in \mathbb{N}$ is a computable map $c : \mathbb{N}^4 \to \mathbb{N}$ which satisfies the following assumptions.[4]

ASSUMPTION 1: *Given any pair (d_s, s), the complexity costs associated with the pair (r, k) are defined if and only if the computation $\{r\}(d_s, s)$ is defined. Formally, we assume that if $c \in \mathbb{N}$ is a Turing machine computing a complexity cost function then*

$$\{c\}(r, k, d_s, s) \downarrow \quad \textit{if and only if} \quad \{r\}(d_s, s) \downarrow \tag{4}$$

The second property of a complexity cost function is that it is always possible to check in a computable way whether the complexity costs associated with a given (r, k, d_s, s) exceed a given value or not. Formally we state

ASSUMPTION 2: *Let c be a Turing machine which computes a complexity cost function. Then there exists a Turing machine m_c (which depends on c) such that, for all $(r, k, d_s, s, y) \in \mathbb{N}^5$ we have*

$$\{m_c\}(r, k, d_s, s, y) = \begin{cases} 0 & \textit{if } \{c\}(r, k, d_s, s) \leq y \\ 1 & \textit{otherwise} \end{cases} \tag{5}$$

The next property of complexity cost functions says that given any arbitrary value y for the complexity cost it is possible to work out the 'stage of the computation' reached at complexity cost y. If the total complexity cost is below y, then the stage of the computation reached by y is simply the result of the computation itself. In the formal assumption which follows, $\theta(\cdot)$ should be thought of as the state of the 'interim result' at stage y, while i_y should be thought of as the 'state of the program' at stage y. We state the following as an assumption, although it can be demonstrated

to be a feature of all standard computational models which give rise to the class of general recursive functions.

ASSUMPTION 3: *If c is a Turing machine computing a complexity cost function, then there exist two total computable functions, $\theta : \mathbb{N}^5 \to \mathbb{N}$ and $f : \mathbb{N}^3 \to \mathbb{N}$ such that $\theta(r, k, d_s, s, 0) = \langle d_s, s\rangle$ for all $(r, k, d_s, s) \in \mathbb{N}^4$, and (set $i_0 = 0$ by convention)*

$$f(r, i_y, \theta(r, k, d_s, s, y)) = \langle \theta(r, k, d_s, s, y + 1), i_{y+1}\rangle \tag{6}$$

and

$$\{c\}(r, k, d_s, s) \leq y \quad \Rightarrow \quad \theta(r, k, d_s, s, y) = \{r\}(d_s, s) \tag{7}$$

We also define f_θ to be the function which computes the first element of the coded pair which is computed by f itself. In other words, $f_\theta(r, i_y, \theta(r, k, d_s, s, y)) = \theta(r, k, d_s, s, y + 1)$.

The last assumption on complexity cost functions simply says that between stages y and $y + 1$ of the computation, only a finite set of alterations to the interim result $\theta(\cdot)$ are possible.

ASSUMPTION 4: *If c is a Turing machine computing a complexity cost function, then for some pair of functions $\theta(\cdot)$ and $f(\cdot)$ as in Assumption 3 there exists a total computable function z such that*

$$f_\theta(r, i_y, \theta) \in \rangle z(\theta) \langle \tag{8}$$

From now on we will refer to a complexity cost function which satisfies Assumptions 1, 2, 3 and 4 simply as an *admissible* complexity cost function.

5. THE CONTRACTING PROBLEM WITH COMPLEXITY COSTS

We are now ready to introduce complexity costs in the contracting problem we have described in Section 2 and analyze their impact on the solution to the problem itself.

We start by modifying the definition of a computable contract that we introduced in Section 3 above. Indeed, when complexity costs are explicitly taken into account

the contracting parties have to agree on both the net share of the surplus each party will receive—the computable sharing rule r—*and* on how to share the complexity costs associated with the contract in question—the vector coded by k. Therefore, we take a computable contract to be the pair of natural numbers $(r, k) \in \mathbb{N}^2$.

We then proceed to modify the general contracting problem (2) in order to incorporate complexity costs. Imagine that the complexity costs can be measured in some resource which trades at a given relative price with the resource which is the object of risk-sharing for the two parties concerned. Call this relative price $q > 0$. Then we modify the general contracting problem by requiring the contracting parties to cover with the surplus available in every given state of nature the complexity costs imputed in the resource the parties are sharing. In other words, we require the parties to subtract the costs, times the relative price q, from the total available surplus for any given state of nature.[5] This clearly implies a restriction on the set of feasible contracts since it introduces a state-by-state upper bound on the cumulative complexity costs that can be associated with each feasible contract.[6]

Intuitively, this way to introduce complexity costs in the parties' risk-sharing problem corresponds to an interpretation of these costs as legal costs. Complexity is not necessarily associated with devising the contract but rather with the writing and enforcement of such a contract. Alternatively, one could interpret the complexity costs as the price, in terms of the resource allocated, of hiring an outsider to devise the contract for the contracting parties. In other words we assume the existence of a market in which such a service is available and may be priced.

The above restriction requires us to modify the feasibility constraint in Problem (2). This constraint needs to be replaced by a constraint that takes into account the complexity costs. Given a Turing machine c computing an admissible complexity cost function and a default allocation of surplus d, a computable contract (r, k) is feasible if and only if

$$r_s^1 + r_s^2 + q\ \{c\}(r, k, s, d_s) \leq d_s^1 + d_s^2. \tag{9}$$

Then the analogue of Problem (2) can be stated as

$$\begin{aligned} \max_{(r,k)\in\mathbb{N}^2} \quad & \mathcal{G}[\{r\}] \\ s.t. \quad & \{r\}(\cdot) \in \mathcal{T} \end{aligned} \tag{10}$$

where $\mathcal{T}$ is the set of pairs $(r,k) \in \mathbb{N}^2$ satisfying:

$$\begin{aligned} \sum_{s=1}^{N} U_1(r_s^1, s)p_s &\geq \sum_{s=1}^{N} U_1(d_s^1, s)p_s \\ \sum_{s=1}^{N} U_2(r_s^2, s)p_s &\geq \sum_{s=1}^{N} U_2(d_s^2, s)p_s \\ r_s^i \geq 0 \quad \forall\, i = 1,2 \quad & \forall\, s \in \mathcal{S} \\ r_s^1 + r_s^2 + q\ \{c\}(r,k,s,d_s) \leq d_s^1 + d_s^2 \quad & \forall\, s \in \mathcal{S}. \end{aligned} \tag{11}$$

We will refer to a Turing machine which solves Problem (10) as an optimal computable contract with complexity costs c.

6. Optimal Contracts With Complexity Costs

In this section we present the main results of our analysis. We start by showing that the parties' choice problem in the presence of complexity costs (Problem (10)) has indeed a solution. We then proceed to give a partial characterization of such an optimal computable contract. We first show that in the class of risk-sharing problems it is possible to find contracting problems such that when complexity is explicitly accounted for, the parties' expected utility corresponding to the optimal computable contract is bounded away from their expected utility corresponding to the first best contract. In other words, the complexity costs have a non negligible impact on the parties' welfare. We can state this result in these terms since we are able to show that such a result holds even in a class of risk-sharing problems yielding a computable first-best contract. Hence, we can be sure that the loss in welfare is not due to the algorithmic restriction imposed on feasible contracts but rather to the complexity costs.

Having proved that the parties' expected utility corresponding to the optimal computable contract is bounded away from the first best, still does not demonstrate that the optimal computable contract is incomplete. To proceed further in this direction,

we add two more restrictions to the class of complexity measures which we termed admissible. We are then able to show that, for any given complexity measure in this marginally more restrictive class it is possible to find a contracting problem such that the optimal choice for the contracting parties, when they take complexity costs into account, is to write no contract and rely on the default allocation of surplus.

We then conclude the analysis by asking whether the optimal contract with complexity costs is constrained efficient even as complexity costs shrink to zero. We show that indeed this is the case: when complexity costs tend to zero the limit optimal contract coincides with the first best allocation of surplus.

Before proceeding any further we prove a preliminary result. Let c be a given admissible complexity cost function. Then, for any state there exists a finite set within which the output of any feasible contract must lie. This finite set can be effectively computed from the state s.

Formally, this preliminary result can be stated as the following lemma.

LEMMA 1: *Let c and d be given. Then there exists a computable function h such that for any feasible contract $(r,k) \in \mathcal{T}$, as defined in (11), we have*

$$\{r\}(s, d_s) \in \rangle h(s) \langle \tag{12}$$

PROOF: Notice that, by (9), $\{c\}(r,k,s,d_s) \leq (d_s^1 + d_s^2)/q$ for every $s \in \mathcal{S}$. We display an informal algorithm for computing the elements of the set $\rangle h(s) \langle$ in (12). Let $\Theta_0 = \langle d_s \rangle$, and define $\Theta_1 = z(\Theta_0)$ where z is the computable function of Assumption 4. Denote by θ_1^e the elements of Θ_1. In other words let $\rangle \Theta_1 \langle = \{\theta_1^1, \cdots, \theta_1^e, \cdots \theta_1^{E_1}\}$. Use now z recursively to define

$$\Theta_n = \left\langle \left[\bigcup_{e=1}^{E_{n-1}} \rangle z(\theta_{n-1}^e) \langle \right] \right\rangle \tag{13}$$

Notice now that by (7) of Assumption 3 and (8) of Assumption 4, we must have

$$\{r\}(s, d_s) \in \rangle \Theta_y \langle \tag{14}$$

whenever $y \geq \{c\}(r,k,s,d_s)$. By feasibility of (r,k), condition (9), the latter inequality, must clearly hold if we set $y \geq (d_s^1 + d_s^2)/q$. Since z is computable by Assumption 4, and the values of Θ_n are obtained by finite recursion starting from the values of s and d_s, clearly there exists a computable function h such that $h(s) = \Theta_y$ for every $s \in \mathcal{S}$, with $y \geq \max_{s\in\mathcal{S}}(d_s^1 + d_s^2)/q$. ■

We now restrict the class of complexity measures we consider so as to require that the default allocation of surplus is the least cost allocation a contract can implement. Indeed, one way to implement such an allocation is to write no contract at all.

ASSUMPTION 5: *The complexity cost function c is such that there always exists a contract (r,k) which computes the default sharing rule and has zero complexity costs for all s in $\mathcal{S}$. In other words c is such that there exists (r,k) such that*

$$\{r\}(s,d_s) = d_s \quad \textit{and} \quad \{c\}(r,k,s,d_s) = 0 \quad \forall\, s \in \mathcal{S} \tag{15}$$

Notice that Assumption 5 guarantees that the feasible set of maximization Problem (10) is not empty. We have now all the elements to prove our first existence result.

PROPOSITION 1: *Given any admissible complexity cost function c, an optimal computable contract with complexity costs c (a solution to Problem (10)) exists.*

PROOF: Notice that whenever two contracts (r,k) and (r',k') are both feasible and are such that $\{r\}(s,d_s) = \{r'\}(s,d_s)$ for all $s \in \mathcal{S}$, then it must be that $\mathcal{G}[\{r\}] = \mathcal{G}[\{r'\}]$. From Lemma 1, we also know that any feasible contract (r,k) satisfies

$$\{r\}(s,d_s) \in \bigcup_{s\in\mathcal{S}} \rangle h(s) \langle \quad \forall s \in \mathcal{S} \tag{16}$$

It then follows immediately that $\mathcal{G}[\cdot]$ can only take on a finite set of different values as we let (r,k) vary across all possible feasible contracts. From Assumption 5 we know also that a feasible contract exists. Therefore Problem (10) has a solution. ■

The next result compares the solutions (in expected utility terms) to the same contracting problem when there are no complexity costs and when complexity costs have been taken into account.

Let an admissible complexity cost function c be given. Consider now a parametric class of contracting problems (obtained for instance by varying the parties' expected utilities indexed by $\lambda \in \Lambda$). Assume that for each $\lambda \in \Lambda$ a unique first best as defined in (2) exists. Let this be denoted by r^*_λ. Assume further that as λ varies in Λ infinitely many (at least a countable infinity) distinct first best sharing rules are obtained. Then the following result holds.

PROPOSITION 2: *Whatever the admissible complexity cost function c there exists at least one $\lambda \in \Lambda$ such that the utility of the optimal contract with complexity costs c is bounded away from the expected utility of the first best r^*_λ, for at least one agent $i = 1, 2$.*

PROOF: From Lemma 1 we know that given c any feasible contract must satisfy $\{r\}(s, d_s) \in \rangle h(s) \langle$. Since as λ varies we obtain infinitely many distinct first best rules r^*_λ, and $\mathcal{S}$ is finite, for at least one $s \in \mathcal{S}$ it must be that $r^{i*}_{s,\lambda}$ takes at least a countable infinity of distinct values as λ varies. Let this set of distinct values be denoted by $\rho(\lambda, s)$.[7] Given the cardinality of the sets it is impossible that $\rho(\lambda, s) \subseteq \rangle h(s) \langle$. Therefore, for some $\lambda \in \Lambda$ it must be the case that any Turing machine which computes the sharing rule r^*_λ is not feasible in Problem (10). ■

The next result is of interest for two distinct reasons. First of all it provides an example of a parametric class of problems to which Proposition 2 applies. Secondly, it clarifies that Proposition 2 applies to some parametric classes of contracting problems which yield a set of *computable* first best sharing rules. For such parametric classes, the difficulty of computing the first best when complexity costs are taken into account is then clearly a result of the complexity costs themselves, rather than of the possible non-computability of the first best sharing rule.

PROPOSITION 3: *There exists a parametric class of co-insurance problems to which Proposition 2 applies and such that the first best sharing rule is computable for all $\lambda \in \Lambda$.*

PROOF: Consider the co-insurance contracting problem of Section 2. Assume that the total surplus is the same in each state of nature, $\pi_s = \pi$ for every $s \in \mathcal{S}$, and the default division of surplus is such that the entire surplus is owned by agent 2 in states $\{1, \ldots, n\}$ and by agent 1 in the remaining states of nature $\{n+1, \ldots, N\}$ where $1 \leq n < N$. We specify the form of the bargaining game by assuming that the contract r is chosen by letting agent 1 make a take-it-or-leave-it offer to agent 2. Agent 2 can accept or reject the offer. He will choose to accept if and only if his 'participation constraint' is met. Finally, we assume that the parties utilities are not state dependent: $U_i(r_s^i, s) = U_i(r_s^i)$ for every $s \in \mathcal{S}$ and every $i = 1, 2$. Problem (2) becomes:

$$\begin{array}{ll} \max\limits_{r} & \sum\limits_{s=1}^{N} U_1(r_s^1) p_s \\ \text{s.t.} & r \in \mathcal{F} \end{array} \tag{17}$$

where $\mathcal{F}$ is defined in (3).

By completely standard arguments the first best contract assigns a constant share of the surplus to both agents. In other words, $r_s^{i*} = r^{i*}$ for every $s \in \mathcal{S}$ and every $i = 1, 2$. Moreover, the participation constraint for agent 2 implicitly defines the first best share of surplus for agent 2:

$$U_2(r^{2*}) = p\, U_2(\pi) + (1 - p)\, U_2(0) \tag{18}$$

where $p \equiv \sum_{h=1}^{n} p_h$. Without loss of generality we can take $U_2(0) = 0$ and $U_2(\pi) = 1$. Let now Λ be the set of rational numbers in the interval $[0, \pi]$, and take $p = U_2(\lambda)$ for a given $\lambda \in \Lambda$. The first best contract is then:

$$r_\lambda^* = (r^{1*} = \pi - \lambda,\ r^{2*} = \lambda) \tag{19}$$

Consider now the parametric class we have just defined when an admissible complexity cost function $c \in \mathbb{N}$ is given. Proposition 2 applies. Indeed, Λ contains a countable infinity of elements, and two distinct values of λ yield two distinct first best contracts. It follows immediately from (19) and the fact that the surplus π is a rational number that the first best is computable for any rational number $\lambda \in \Lambda$. ■

We shall now introduce a further assumption on complexity measures which we denote 'no free transfer' condition. We assume that the only contracts which have zero complexity costs for all states of nature are those contracts which leave the default division of surplus unaffected. Formally

ASSUMPTION 6: *For any* (r, k)

$$\{c\}(r, k, s, d_s) = 0 \ \forall \ s \in \mathcal{S} \quad \Rightarrow \quad \{r\}(s, d_s) = d_s \ \forall \ \in \mathcal{S}. \tag{20}$$

We have now all the elements to prove our next result. Intuitively, for any complexity measure in the class of measures that satisfy Assumptions 1-6 it is possible to think of a contracting problem (in particular of a default allocation of surplus) such that the best option left to the contracting parties when complexity costs are taken into account is not to write any contract at all. Moreover, such default allocation of surplus differs from the first best contract.

PROPOSITION 4: *Let* $\mathcal{G}$ *and* c *be given. Then there exists a default division of surplus which is different from the first best contract and is such that the optimal contract with complexity costs which solves Problem (10) is equal to such default.*

PROOF: Consider any contracting problem in which the default division of surplus, d, does not coincide with the first best. Let now (r_d, k_d) be the optimal contract with complexity costs, given d. If $r_d = d$ there is clearly nothing more to prove. Assume therefore that $r_d \neq d$. Let r_d^* be the first best contract given d.

Since $r_d \neq d$, it follows from Assumption 6 that there exists a state of nature $s \in \mathcal{S}$ such that

$$\{c\}(r_d, k_d, s, d_s) \geq 1 \tag{21}$$

From (21), it then follows that there exists a $\xi > 0$ (independent of d) such that

$$\mathcal{G}[r_d^*] - \mathcal{G}[\{r_d\}] > \xi \tag{22}$$

Notice now that, using the continuity of $\mathcal{G}$ and of $U_i(\cdot, s)$ for all s and for all i, given any $\gamma > 0$ it is possible to find a d such that

$$\mathcal{G}[r_d^*] - \mathcal{G}[d] < \gamma \tag{23}$$

Since ξ on the right-hand side of (22) is independent of d, we can now conclude from (22) and (23) that for γ sufficiently small it must be the case that

$$\mathcal{G}[d] > \mathcal{G}[\{r_d\}] \tag{24}$$

Clearly, (24) contradicts the fact that $r_d \neq d$ and therefore the proof of the proposition is complete. ■

There is a sense in which Proposition 4 is not surprising. This is apparent once we observe that any complexity cost function satisfying Assumption 6 is such that any contract that implements an allocation of surplus different from the default is associated with a *fixed cost*. In other words there is a fixed cost which the parties have to pay to modify the default with an explicit contract.

If we now parameterize the contracting problem with the default allocation of surplus, we can show that by choosing a default allocation 'close' to the first best we can make sure that for any size of the contractual fixed cost it is not worth for the contracting parties to pay the fixed costs and modify the default allocation, which can be implemented at zero complexity costs.

A symmetric result may now be proved. If instead of varying the default allocation in Problem (10) we vary the relative price of the complexity measure, we are able to show that for every given default contracting problem in which complexity costs are accounted for, it is possible to find a level of the relative price such that it is not worth for the parties to write any contract and modify the default allocation of surplus.

Consider a contracting problem with a given complexity cost function c and take the default division of surplus d to be given. Let such a problem be parameterized by the relative price q as in (11). In this parametric class of contracting problems the following result holds.

PROPOSITION 5: *Let $\mathcal{G}$ and c be given. There exists a $q > 0$ such that the optimal*

contract with complexity costs is equal to the given default and it is different from the first best contract.

PROOF: Let r_d be a contract which prescribes shares of the surplus precisely equal to the default and which yields zero complexity costs in any state.

From (9) and (21) we can clearly choose q large enough so as to ensure that for any r which does not prescribe exactly the default is not feasible. ■

We conclude the characterization of optimal co-insurance contracts with complexity costs by analyzing how these contracts change as the size of the complexity costs becomes arbitrarily small. In particular we analyze the solution to Problem (10) as the relative price q becomes infinitesimal. We obtain that the optimal contract with complexity costs converges to the first best contract solving Problem (2). 'Continuity at the limit' holds in our model.

For a given complexity cost function c and default division of surplus d consider the class of contracting problems parameterized by q and denote $(r_q^*, k_q^*) \in \mathbb{N}^2$ the solution to Problem (10) for any given q. Recall that r^* denotes the first best contract solving Problem (2). The following result holds.

PROPOSITION 6: *Let $\mathcal{G}$ and c be given. Given any $\varepsilon > 0$ there always exists a relative price q and an optimal computable contract with complexity costs r_q^* such that*

$$\left|\mathcal{G}[r^*] - \mathcal{G}[\{r_q^*\}]\right| < \varepsilon \tag{25}$$

PROOF: Consider party i's expected utility associated with the optimal contract with complexity costs (r_q^*, k_q^*):

$$\sum_{s=1}^{N} U_i(r_{s,q}^{i*}, s) p_s \tag{26}$$

where $(r_{s,q}^{1*}, r_{s,q}^{2*}) = \{r_q^*\}(s, d_s)$. Expected utility (26) is bounded below by the expected utility

$$\sum_{s=1}^{N} U_i[(r_s^{i*} - q\ k_{s,q}^{i*}\ y), s] p_s \tag{27}$$

where $y = \{c\}(r_q^*, k_q^*, s, d_s)$. Expected utility (26) is also bounded above by the first best expected utility

$$\sum_{s=1}^{N} U_i(r_s^{i*}, s)p_s \tag{28}$$

where r_s^{i*} is the i's first best share in state of nature s when complexity costs are not considered. Notice that (27) is monotonically decreasing in q while (28) does not depend on q. We can therefore conclude that continuity of the mapping $\mathcal{G}[\cdot]$ implies (25). ■

Therefore, the optimal contract with complexity costs remains constrained efficient in the limit as q tends to zero.

7. CONCLUSIONS

In this paper we have provided a framework to analyze the impact of general complexity costs associated with the writing and the implementation of contracts on the optimal choice of the contracting parties. In particular, we conclude that for any complexity measure in a very general class there exist situations in which, because of complexity considerations, it is optimal for the contracting parties to rely on the default division of surplus rather than try to modify it by means of an explicit contract. In other words to write an incomplete contract. However, in the absence of any strategic role for complexity costs, the incompleteness we obtain is constrained efficient.

The framework we consider may be made more specific to obtain the characterization of the optimal incomplete contract in a number of possible ways. In particular we think that an interesting issue to explore is whether it is possible to derive a theory of standardization of contracts using the framework we have developed. In the Introduction we highlighted the distinction between ex-ante complexity costs, which are associated with the writing of the contract and ex-post costs which are associated with the costs of identifying the realized state of nature, computing and implementing the prescriptions of the contract. The costs of computing the outcome (the net trades) prescribed by the contract, however, can be interpreted as either ex-ante or ex-post (or possibly both). If all computations are carried out ex-ante, the contract

can be viewed as containing a 'look-up table' which gives a prescribed net trade for every possible state of nature. Alternatively, the contract can provide a set of rules that the enforcing agency is required to follow to compute the net trades prescribed by the contract *after* the state is realized. Whether the contracting parties will choose the first or the second alternative will depend on the situation in which they operate.

If we denote with $g(s)$ the costs associated with computing the net trades prescribed by the contract when state s is realized it is natural to think that the costs we are considering will affect the parties' expected utilities as costs that depend on the realized state of nature if they are faced only ex-post, at the implementation stage. They will enter only as fixed costs if they are faced ex-ante at the stage in which the contract is written. A natural upper-bound on these fixed costs will be provided by the sum of the costs of computing the outcome for every state of nature: $\sum_{s\in\mathcal{S}} g(s)$. Notice that in a setting in which the contracting parties are risk neutral it seems likely that they will choose to face these costs ex-post since $\sum_{s\in\mathcal{S}} g(s)p_s < \sum_{s\in\mathcal{S}} g(s)$. This will not necessarily be the case if the parties are risk averse. Notice that in the case the parties are infinitely risk averse (and the complexity costs vary across states) they will always choose to face the fixed costs ex-ante.

Finally, it is worth noticing that in a setting like the one just described there may exist an opportunity for a legal agency to provide contracting parties with pre-specified tables which associate outcomes to each state of nature; in a word with standardized contracts. Whether this opportunity is profitable will clearly depend on how common the contracting situation is. In particular it may become relevant whether there exists a whole set of situations which are 'close' but not identical to the contracting situation for which the standardized contract is available. Clearly this will be a situation in which complexity considerations may lead to contractual incompleteness of a less extreme form than the one we derive in the paper.

NOTES

1. We revise this distinction to some extent in the Conclusions when we touch on the issue of 'standardized' contracts.

2. We denote the set of rational numbers with the symbol $\mathcal{Q}$.

3. See Proposition 1 of Anderlini and Felli (1994). For an explicit treatment of the problem with a countable state space, see also Proposition 1 of Anderlini and Felli (1993).

4. The first two assumptions below are well known in the mathematical literature on recursive function theory. They are often referred to as the Blum (1967) axioms for dynamic complexity measures. The last two assumptions, in particular Assumption 4, are a further restriction which we impose on the class of complexity cost functions which we consider here.

5. An alternative way to proceed would have been to impute the complexity costs to the parties as a direct 'out-of-pocket' expense, either in terms of utility or in terms of the resource which is the object of the risk-sharing agreement. All the results which we report below would still be valid in these cases.

6. In general we could have formulated the parties' problem with complexity costs using a state-by-state arbitrary upper-bound on the overall amount of complexity costs that it is feasible for a contract to generate. All our result in Section 6 would still hold.

7. Recall that when complexity costs are not accounted for, Problems (2), the set of first best sharing rules $\rho(\lambda, s)$ is independent of d_s.

REFERENCES

Abreu, D. and Rubinstein, A.: 1988, The Structure of Nash Equilibrium in Repeated Games with Finite Automata, *Econometrica* **56**, 1259–81.

Aghion, P. and Tirole, J.: 1997, Formal and Real Authority in Organizations, *Journal of Political Economy,* forthcoming.

Aghion, P., Dewatripont, M. and Rey, P.: 1994, Renegotiation Design with Unverifiable Information, *Econometrica* **62**, 257–82.

Anderlini, L.: 1989, Some Notes on Church's Thesis and the Theory of Games, *Theory and Decision* **29**, 19–52.

Anderlini, L. and Felli, L.: 1993, Incomplete Written Contracts: Undescribable Sates of Nature, *Theoretical Economics Discussion Paper TE/93/263*, STICERD, London School of Economics.

Anderlini, L. and Felli, L.: 1994, Incomplete Written Contracts: Undescribable Sates of Nature, *Quarterly Journal of Economics* **109**, 1085–1124.

Anderlini, L. and Felli, L.: 1997, Costly Coasian Contracts, University of Cambridge, mimeo.

Blum, M.: 1967, A Machine-Independent Theory of the Complexity of Recursive Functions, *Journal of the Association of Computing Machinery* **14**, 322–336.

Chung, T. Y.: 1991, Incomplete Contracts, Specific Investments, and Risk Sharing, *Review of Economic Studies* **58**, 1031–42.

Cutland, N. J.: 1980, *Computability: An Introduction to Recursive Function Theory*, Cambridge University Press, Cambridge.

Dye, R. A.: 1985, Costly Contract Contingencies, *International Economic Review* **26**, 233–50.

Grossman, S. J. and Hart, O. D.: 1986, The Costs and Benefits of Ownership: A Theory of Vertical and Lateral Integration, *Journal of Political Economy* **94**, 691–719.

Hart, O. D. and Holmström, B.: 1987, The Theory of Contracts, *in* T. F. Bewley (ed.), *Advances in Economic Theory, Fifth World Congress*, (Cambridge University Press, Cambridge).

Hart, O. D. and Moore, J.: 1988, Incomplete Contracts and Renegotiation, *Econometrica* **56**, 755–85.

Hart, O. D. and Moore, J.: 1990, Property Rights and the Nature of the Firm, *Journal of Political Economy* **98**, 1119–58.

Maskin, E. and Tirole, J.: 1996, Dynamic Programming, Unforseen Contingencies, and Incomplete Contracts, Harvard University, mimeo.

Nöldeke, G. and Schmidt, K. M.: 1995, Option Contracts and Renegotiation: A Solution to the Hold-Up Problem, *RAND Journal of Economics* **26**, 163–179.

Piccione, M.: 1992, Finite Automata Equilibria with Discounting, *Journal of Economic Theory* **56**, 180–193.

Rogers, H.: 1967, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill Book Company, London.

Rubinstein, A.: 1986, Finite Automata Play the Repeated Prisoner's Dilemma, *Journal of Economic Theory* **39**, 83–96.

Rubinstein, A. and Piccione, M.: 1993, Finite Automata Play a Repeated Extensive Game, *Journal of Economic Theory* **61**, 160–168.

Segal, I.: 1995, Complexity and Renegotiation: A Foundation for Incomplete Contracts, Harvard University, mimeo.

Tirole, J.: 1994, Incomplete Contracts: Where do we Stand?, *Technical report*, IDEI, Toulouse.